Analysis Methods in Atmospheric and Oceanic Science

AOSC 652

Root Finding and Function Minimization

Week 7, Day 1

10 Oct 2016

AOSC 652: Analysis Methods in AOSC

Student projects:

- 20% of the final grade: you will receive a numerical score for the project and final grade will be found via:

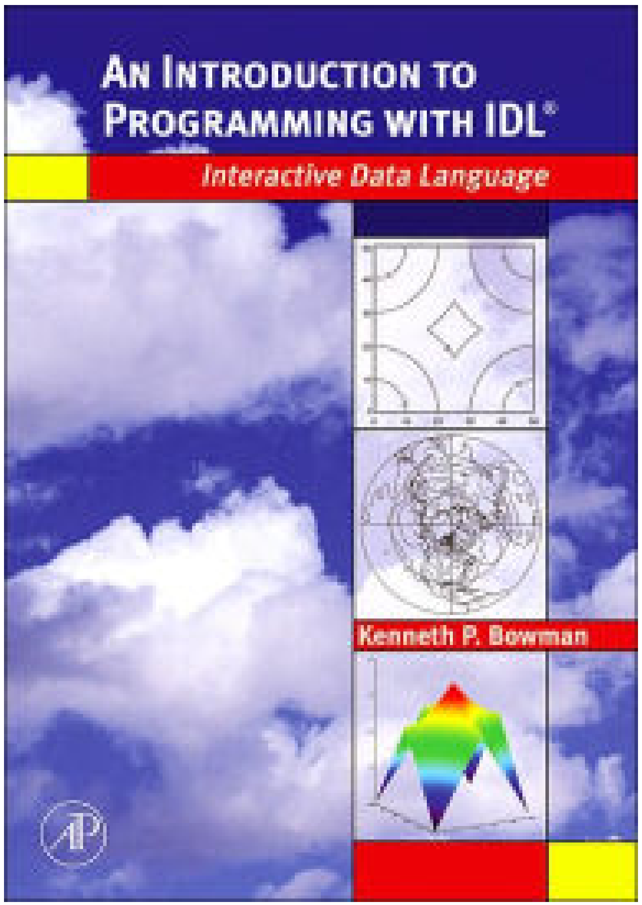
  Final Grade = (0.1)×(Attendance + Participation) + (0.7)×(Homework) + (0.2)×(Final Project)

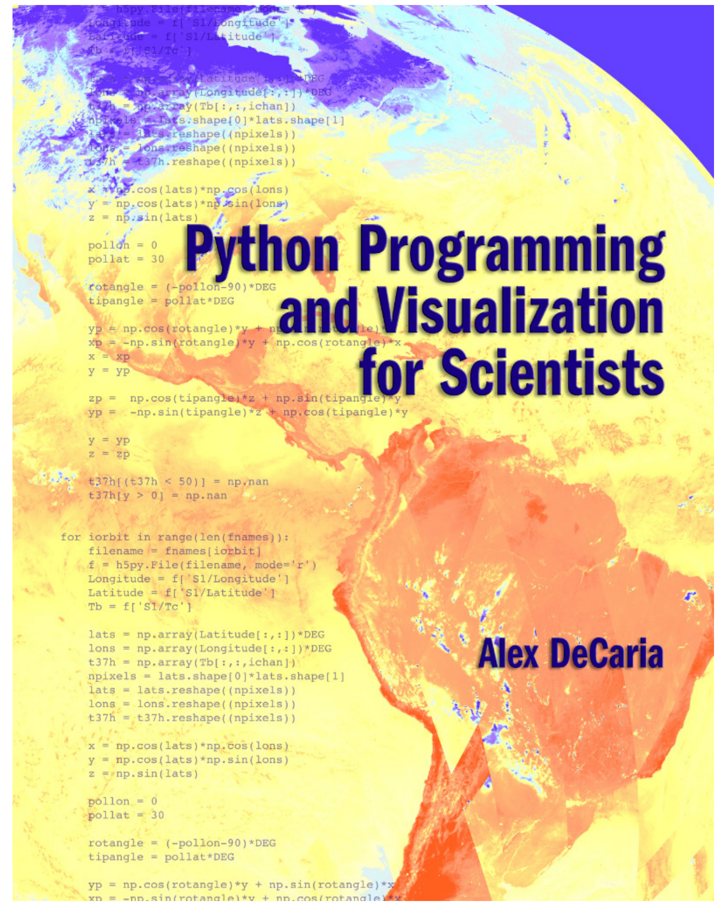
- 28, 30 Nov , 2, 5, and 7 Dec set aside for “in class” work on your project
- Thurs 8 & Fri 9 Dec: *students present their project (10 minute talks)*, prepared using either Powerpoint, Open Office, etc and converted to PDF prior to the start of class
- Each student must turn in a *brief* written description of the project as well as all *code* used to complete the project
- Good to begin thinking about your project: application of techniques learned in class to a ***scientific problem of your interest***
- I am available to discuss potential projects at any time

AOSC 652: Analysis Methods in AOSC

Modular Tracts:

- Next week, Python will meet here, and IDL will meet in CSS 3408
- Please order the appropriate book ASAP





Root Finding and Function Minimization

Solution of many scientific problems requires finding either the root or the minimum of an equation (or set of equations) which can be highly non-linear:

What is meant by root?

Root Finding and Function Minimization

Solution of many scientific problems requires finding either the root or the minimum of an equation (or set of equations) which can be highly non-linear:

Roots of quadratic equation:

$$ax^2 + bx + c = 0$$

Roots are:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Root Finding and Function Minimization

Solution of many scientific problems requires finding either the root or the minimum of an equation (or set of equations) which can be highly non-linear:

Roots of quadratic equation:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Roots of cubic equation ?

Root Finding and Function Minimization

Solution of many scientific problems requires finding either the root or the minimum of an equation (or set of equations) which can be highly non-linear:

Roots of quadratic equation:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Roots of cubic equation:

Can be expressed in closed form, but very few people can state the solution from memory ☺

http://en.wikipedia.org/wiki/Cubic_function

Root Finding and Function Minimization

Solution of many scientific problems requires finding either the root or the minimum of an equation (or set of equations) which can be highly non-linear:

Roots of quadratic equation:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Higher order polynomials?

Root Finding and Function Minimization

Solution of many scientific problems requires finding either the root or the minimum of an equation (or set of equations) which can be highly non-linear:

Roots of quadratic equation:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Even an equation as simple as $cos(x) - x^2$ **= *0*** ***cannot*** be solved analytically!

Root Finding and Function Minimization

Method #1: Direct Search

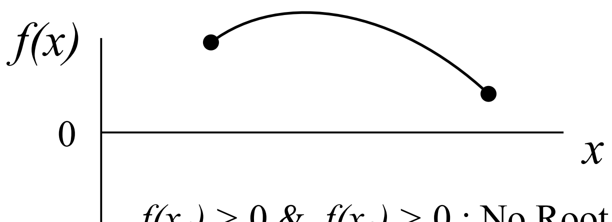
- Specify interval, x_1 to x_2, where root(s) is assumed (or known) to occur
- Divide x_1 to x_2 into various sub-intervals, spaced by a distance $x_{Tolerance}$ that represents the accuracy of the root(s)
- Search through each interval to see where root(s) is (are) present

AOSC 652: Analysis Methods in AOSC

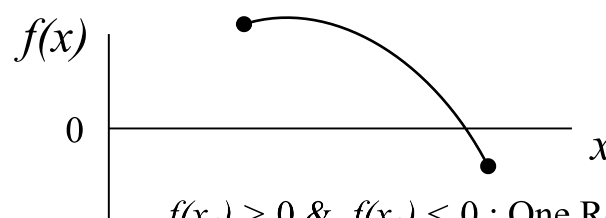
Root Finding and Function Minimization

Method #1: Direct Search

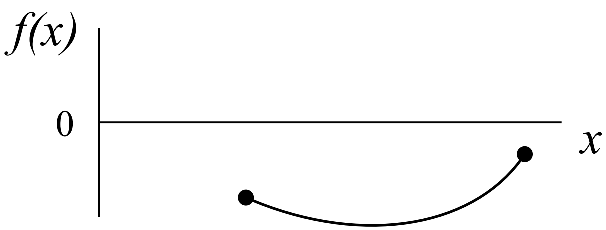
- Specify interval, x_1 to x_2, where root(s) is assumed (or known) to occur
- Divide x_1 to x_2 into various sub-intervals, spaced by a distance $x_{Tolerance}$ that represents the accuracy of the root(s)
- Search through each interval to see where root(s) is (are) present



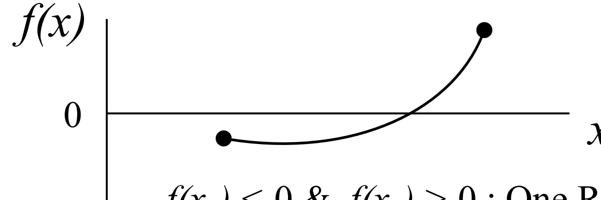
$f(x_1) > 0$ & $f(x_2) > 0$: No Root between x_1 and x_2



$f(x_1) > 0$ & $f(x_2) < 0$: One Root between x_1 and x_2



$f(x_1) < 0$ & $f(x_2) < 0$: No Root between x_1 and x_2



$f(x_1) < 0$ & $f(x_2) > 0$: One Root between x_1 and x_2

Root Finding and Function Minimization

Method #1: Direct Search

- Many “pathological” situations can occur
  - some related to spacing not being fine enough
  - some related to inherent behavior of function

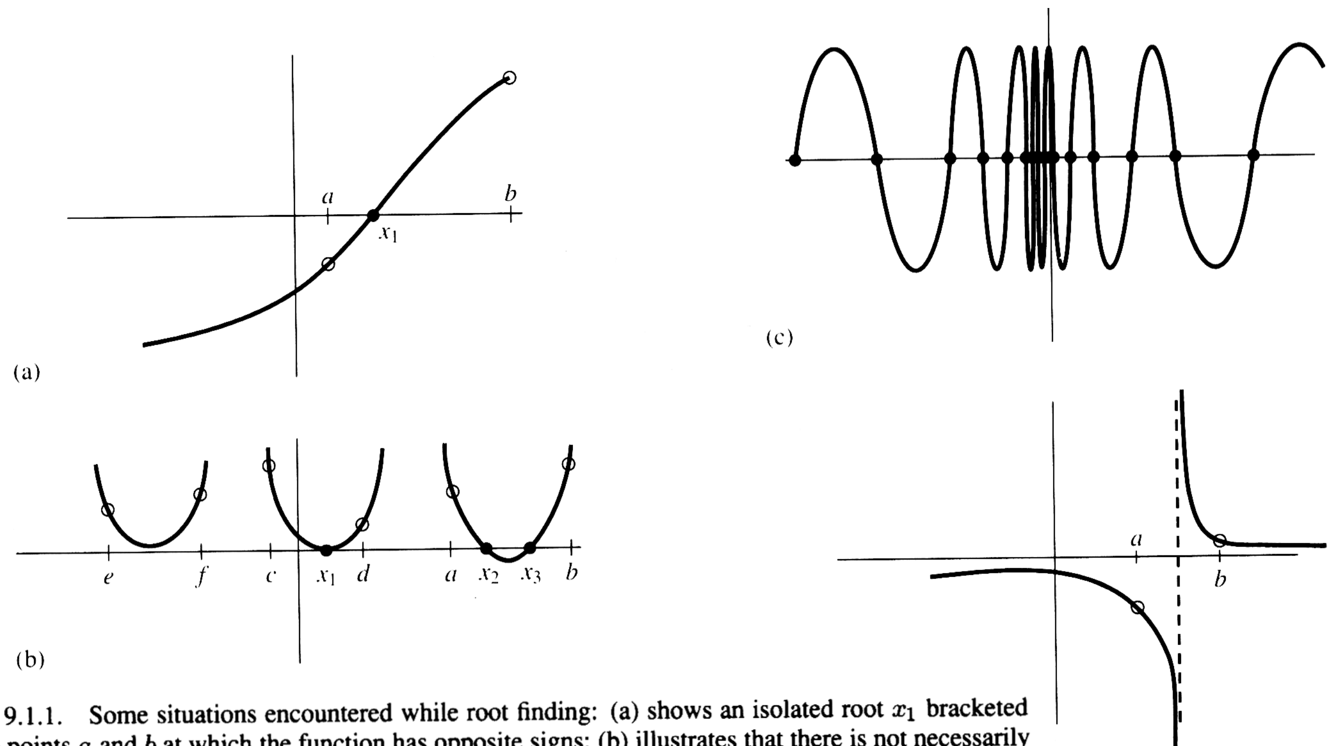


Figure 9.1.1. Some situations encountered while root finding: (a) shows an isolated root x_1 bracketed by two points a and b at which the function has opposite signs; (b) illustrates that there is not necessarily a sign change in the function near a double root (in fact, there is not necessarily a root!); (c) is a pathological function with many roots; in (d) the function has opposite signs at points a and b, but the points bracket a singularity, not a root.

Press *et al.*, Numerical Recipes

AOSC 652: Analysis Methods in AOSC

Root Finding and Function Minimization

Method #1: Direct Search

- Let's apply the direct search method to the function $cos(x) - x^2 = 0$

AOSC 652: Analysis Methods in AOSC

Root Finding and Function Minimization

Method #1: Direct Search

- Let's apply the direct search method to the function $cos(x) - x^2$ **= *0***
- Exercise: plot $cos(x) - x^2$, from x = 0 to 1, and ascertain:
  - number of roots between 0 and 1
  - approximate value of root(s)

    Note: treat "*x*" as radians

Root Finding and Function Minimization

Method #1: Direct Search

- Let's apply the direct search method to the function $cos(x) - x^2 = 0$
- Let's find the root using "direct search"
  - copy file ~rjs/aosc652/week_07/direct_search.f to your work area
  - have a look at the code (handout); then, compile and run
  - find the root to an accuracy of 1.E–2
  - repeat for accuracy of 1.E–4

AOSC 652: Analysis Methods in AOSC

Root Finding and Function Minimization

Method #1: Direct Search

- Let's apply the direct search method to the function $cos(x)-x^2$ **= *0***
- Let's find the root using "direct search"
  - – copy file ~rjs/aosc652/week_07/direct_search.f to your work area
  - – have a look at the code (handout); then, compile and run
  - – find the root to an accuracy of 1.E–2
  - – repeat for accuracy of 1.E–4

How many times was the function called?

How long would this process have taken if the function, rather than $cos(x)-x^2$, was a multiple scattering, radiative transfer calculation that took ~2 mins to converge (for each call)?

Method #2: Bisection Method

Advancement on the direct search method that is applicable when only one root is known to occur within a specified interval

1. Evaluate function at x_S, x_E, and midpoint x_M : $f(x_S)$, $f(x_E)$, and $f(x_M)$
2. If $f(x_S) \cdot f(x_M) > 0$ and $f(x_M) \cdot f(x_E) < 0$, root must lie between x_M and x_E (i.e., root lies within interval for which product is negative)

$f(x_S) \cdot f(x_M) > 0$

$f(x_M) \cdot f(x_S) < 0$

Root lies between x_M and x_S

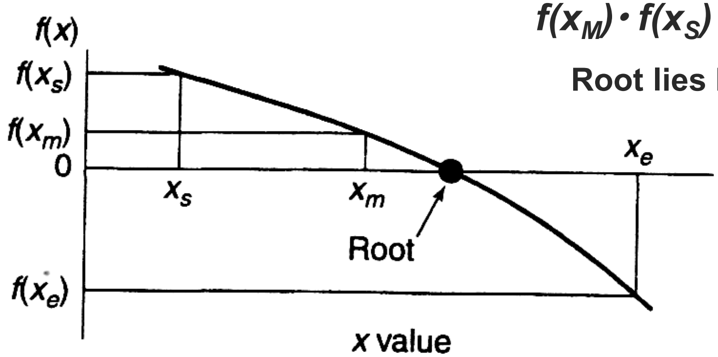


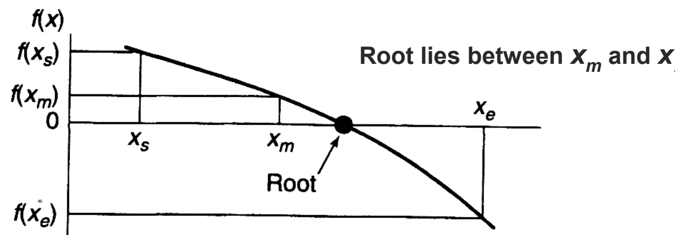
Figure 4-6 Schematic of Bisection Method

AOSC 652: Analysis Methods in AOSC

Method #2: Bisection Method

Advancement on the direct search method that is applicable when only one root is known to occur within a specified interval

1. Evaluate function at x_S, x_E, and midpoint x_M : $f(x_S)$, $f(x_E)$, and $f(x_E)$
2. If $f(x_S) \cdot f(x_M) > 0$ and $f(x_M) \cdot f(x_E) < 0$, root must lie between x_M and x_E (i.e., root lies within interval for which product is negative)



3. x_M is used as estimate of root *for this iteration*
4. Convergence if:
   a) $f(x_M)$ = 0 or $f(x_M)$ lies within a certain distance of 0 –or–
   b) update to $|x_M| < x_{Tolerance}$
5. If convergence criteria *not met*:
   a) let end points of the half-interval in which root is located be end points of new interval
   b) return to function evaluation (#1) above

AOSC 652: Analysis Methods in AOSC

Root Finding and Function Minimization

Method #2: Bisection Method

- Copy file ~rjs/aosc652/week_07/bisection.f to your work area
- Have a look at the code (handout); then, compile and run
- Find the root to an accuracy of 1.E–2
- Repeat for accuracy of 1.E–4

Root Finding and Function Minimization

Method #2: Bisection Method

- Copy file ~rjs/aosc652/week_07/bisection.f to your work area
- Have a look at the code (handout); then, compile and run
- Find the root to an accuracy of 1.E–2
- Repeat for accuracy of 1.E–4
  - How many calls to function were required to find the root for each call?

NOTE: when finding roots or minimizing functions, always important to place a limit on the number of iterations and to pay attention to the “tolerance” (accuracy) of the result

How is the number of iterations “limited” in code bisection.f ?

Root Finding and Function Minimization

Method #2: Bisection Method

- Copy file ~rjs/aosc652/week_07/bisection.f to your work area
- Have a look at the code (handout); then, compile and run
- Find the root to an accuracy of 1.E–2
- Repeat for accuracy of 1.E–4
  - How many calls to function were required to find the root for each call?
  - Convergence behavior of the bisection method is not bad (for the chosen function), but the bisection method is rarely used in modern scientific applications

Root Finding and Function Minimization

Method #2: Bisection Method

- Copy file ~rjs/aosc652/week_07/bisection.f to your work area
- Have a look at the code (handout); then, compile and run
- Find the root to an accuracy of 1.E–2
- Repeat for accuracy of 1.E–4
  - How many calls to function were required to find the root for each call?
  - Convergence behavior of the bisection method is not bad (for the chosen function), but the bisection method is rarely used in modern scientific applications
  - How can we use "calculus" to help us find the root?

Root Finding and Function Minimization

Method #3: Newton-Raphson

- Let's use "calculus" to help us find the root:
- Taylor series expansion of a function *f* evaluated near point *x*

$$f(x+\delta) \approx f(x) + f'(x)\,\delta + \frac{f''(x)}{2}\delta^2 + \ldots$$

Root Finding and Function Minimization

Method #3: Newton-Raphson

- Let's use "calculus" to help us find the root:
- Taylor series expansion of a function *f* evaluated near point *x*

$$f(x+\delta) \approx f(x) + f'(x)\,\delta + \frac{f''(x)}{2}\delta^2 + \ldots$$

- Let's convert this "calculus" to a form that will help us find the root of *f(x)* :

$$f(x_1) \approx f(x_0) + f'(x_0)\,\Delta x$$

or

$$f(x_1) \approx f(x_0) + f'(x_0)\,(x_1 - x_0)$$

Root Finding and Function Minimization

Method #3: Newton-Raphson

- Let's use "calculus" to help us find the root:
- Taylor series expansion of a function *f* evaluated near point *x*

$$f(x+\delta) \approx f(x) + f'(x)\,\delta + \frac{f''(x)}{2}\delta^2 + \ldots$$

- Let's convert this "calculus" to a form that will help us find the root of *f(x)* :

$$f(x_1) \approx f(x_0) + f'(x_0)\,\Delta x$$

or

$$f(x_1) \approx f(x_0) + f'(x_0)\,(x_1 - x_0)$$

Of course:

$$f'(x_0) = \left.\frac{df}{dx}\right|_{x_0} \quad \text{the first derivative of } f \text{ evaluated at } x_0$$

Root Finding and Function Minimization

Method #3: Newton-Raphson

- Root is place where *f(x)* = 0

$$0 \approx f(x_0) + f'(x_0)\,\Delta x = f(x_0) + f'(x_0)\,(x_1 - x_0)$$

- If the derivative is evaluated at x_0 then the "next" value x_1 is found by:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

- Convergence if either:
  - $f(x_i) = 0$ or $f(x_i)$ lies within a certain distance of 0 –or–
  - update to $|x_i| < x_{Tolerance}$

- If not converged, next value x_{i+1} is found from:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Root Finding and Function Minimization

Method #3: Newton-Raphson

- Very powerful convergence method provided:
  - derivative is continuous and non-zero near root
  - initial guess for root is “close” to root
  - should use with iteration counter and check to see if root (or function minima) was indeed found
- Can read about Newton Raphson iteration on the web:
  http://en.wikipedia.org/wiki/Newton-Raphson_method

AOSC 652: Analysis Methods in AOSC

Root Finding and Function Minimization

Method #3: Newton-Raphson

- Very powerful convergence method provided:
  - derivative is continuous and non-zero near root
  - initial guess for root is "close" to root
  - should use with iteration counter and check to see if root (or function minima) was indeed found
- Can read about Newton Raphson iteration on the web:
  http://en.wikipedia.org/wiki/Newton-Raphson_method

Numerical algorithm semantics:

- **Newton-Raphson or Newton's method: derivative can be found analytically**
- **Secant method: derivative can only be found numerically**

AOSC 652: Analysis Methods in AOSC

Root Finding and Function Minimization

Method #3: Newton-Raphson

- **Between now and Wednesday**
- Copy file ~rjs/aosc652/week_07/newton_raphson.f
  into your work area
- Have a look at the code and compare / contrast to
  subroutine rtnewt on page 358 of Press *et al.*

Between now and Wed: please look at this code and compare / contrast to the Press subroutine